Physical Layer

TCP/IP class
physical layer

◆ intro - hw concepts
  - topology
  - wan versus lan
  - switches, circuit and packet
◆ ethernet
◆ point to point serial
◆ odds and ends
  - mtu/path mtu/localhost
  - repeaters/bridges/routers
Two basic ideas:
- The link layer can **broadcast** (multicast)
- The link layer is **point to point**, can’t bcast

Other topologies built out of these building blocks

Point/point often **Wide Area Network (WAN)**
- (telcos - equipment is leased)

Broadcast often **Local Area Network (LAN)**
- (enterprise - equipment is owned)
point to point

ring, ring! yadda, yadda!

note: telco network in-between (not Internet)

Jim Binkley
point to point, examples

- modems (POTS/analog)
- ISDN (digital phone)
- RS-232 cable between two computers
- most WAN topologies (not all)
  - T1/T3, T1 classically 23 64k PCM voice lines
- may have “dynamic connections” and need addresses (phone #s), may not (serial cable)

Jim Binkley
broadcast

1 write - many reads in parallel

Jim Binkley
broadcast

- includes one to one
- **broadcast** means 1 to all stations
- **multicast** means 1 to many, includes 1-1, 1-all (broadcast is subset of multicast)
- Examples include ethernet, token-ring, radio
- questions include: can it do CSMA, CD (later)?
- also notion of **multipoint** - simulation of broadcast by 1 to N point to point connections

Jim Binkley
derived topologies

Star

examples:
enet hubs, ATM
derived topologies

**Ring**

examples:
- token ring, fddi

Jim Binkley
derived topologies

Mesh

examples:
Inet backbone

redundancy,
consider A to E

Jim Binkley
WAN vs LAN

◆ 3 kinds of network
  – in terms of geography, ownership, speed
  – 1. WAN - wide area, telcos own equipment point to point
  – 2. MAN - metro area, telcos own, but has broadcast (fddi, SMDS, atm?) (shared?)
  – 3. LAN - ethernet, token-ring, local, enterprise-owned
WANS

- telcos own, operate
- Bellcore, US West, GTE, other RBOCs
- Sprint, MCI too
- European PTTs (Post, Telephone, Telegraph) - monopolies
- folks who brought us ISO/OSI and are trying to bring us ATM
WAN vs LAN

- different cultures, people, technologies, lingo (can you say pleisochronous?)
- WAN focus traditionally on voice, LAN on data
- WAN standardization efforts slow, LAN relatively fast
- somebody who knows both is rare person
WAN characteristics

- focus on voice/low-speed **isochronous** xfer
- customer *rents* equipment and usage from telco
- in past slower than LAN, may change with ATM (maybe not ... 1G enet)
- point to point (connect first, then switch)
WAN examples

- modem over analog phone (POTS)
  - 1200 baud to 28.8k baud (2-3k bps), now 56k?
  - modems can compress, do error correction
- ISDN (some places) - 64k/128k
- leased line/frame relay, 56k to T1 speeds
- STM - synchronous transfer mode
  - T1 - 1.544 megabits per sec, T3 - 44 mbps
- analog/digital cellular wireless (1-2k bps), up to T3 speeds in some cases for pt/pt radio

Jim Binkley
WAN futures

- cable tv - “upstream” has been problem
- ATM as PVC (permanent virtual circuit)
  - OC3 is 155Mbs
  - OC12 is 622Mbs
  - slower/faster possible too, 1G mbps?
  - short term: ATM is T1/T3 replacement
  - long term: might be LAN technology too
- satellite/radio? TBD
Lan examples (all broadcast)

- **Ethernet**
  - 10/100 (switched/full-duplex)/1000/10000?
    - many wiring models so far
    - 1000 is man technology too (5..100 or so km)

- **Token-ring**
  - 16mbps, 100 exists, prognosis not good (see above)

- **FDDI, man, ring, 100 mbps**

- **wireless radio, 1-10 mbps, 802.11 standard**
  - Lucent IEEE wavelan 2-? mbps, 400-800 foot cell?
switches, circuit OR packet

- **circuit switch** - telco voice routing
  - point/point “virtual circuit”
  - connect-time sets up path from end to end
  - **pros:**
    » endpoints don’t need to worry about load, they have path/circuit capacity reserved
    » faster than packet-switch (?)
  - **cons:**
    » circuit wasted if no data
    » if switch crashes, must reconnect
circuit switch - diagram

1. connect protocol
2. send data
3. disconnect

switch

switch

switch

(not in virtual circuit)

switches contain state: (I(n), O(n))

Jim Binkley
packet switch - router

- packet switches used by computers, send data in discrete packets, each packet has addresses
- no connect/disconnect
- each packet is instantaneously routed (output i/f is determined) acc. to table lookup of dest address
  - f(pkt dst, routing table) -> output port
  - routing table may change from pkt to pkt
- pros:
  - good for bursty traffic
  - robust as fate sharing is minimized
packet switches, continued

◆ cons:
  – switches deemed to be faster, since routing table lookup is network layer/sw decision
  – router software can cause warts...
    » “you! set BGP-4 up on that there router ...!”
  – open problem as to how to do isochronous data xfer
fate-sharing (is a bad thing)

- from very high-level POV
- A-E (end to end) is better than A-B-C-D-E in terms of reliability
- if router C goes down in connection framework, A and E are hosed
- if router C goes down in packet switch network, may have delay (reboot) or alternate path
  BUT THE CONNECTION STAYS UP! ....
- fundamental design decision for Internet routing
**ethernet switch means what?**

- ethernet switch - bridge with fast backplane
  - e.g., 8 ports -> 80mbps (8 * 10mbits)/2
  - **star** topology, still support broadcast but
    » we have features, full-duplex (no collisions)
  - can give each end-node its own 10 mbps to another end-node on switch (point/point)
tcp/ip Point of View for WAN

- **sub-net** versus **peer** addressing models
  - sub-net, means we put you in a link-layer box and run on top of you
  - peer - can address all endpoints
  - Internet Protocol (ip) and routers may sit on top of TELCO circuit-switch network (modems/ISDN), examples
    » Inet in WAN, uses T1/T3
    » end user with modem and PPP/SLIP protocols

Jim Binkley
Telco in a TCP box

your computer at home:

- www browser
- tcp transport
- ip network layer
- slip/ppp
- ISDN phone
- sub-network

you don’t send IP packets to phone #s directly

telco cloud

Jim Binkley
Ethernet - intro

- invented at Xerox Parc in early 70’s
- standardized by Dec/Intel/Xerox (DIX)
- signals on cable called the “ether”
- 80% speed of light
- number of different wire types
- doesn’t load as well as token ring, but still cheaper
# ethernet wiring types

<table>
<thead>
<tr>
<th>cable type</th>
<th>alias</th>
<th>connector</th>
<th>length</th>
</tr>
</thead>
<tbody>
<tr>
<td>10BASE5 50ohmRG-11</td>
<td>thicknet</td>
<td>N-type</td>
<td>5*500M</td>
</tr>
<tr>
<td>10BASE2 50ohmRG-58</td>
<td>thinnet</td>
<td>BNC</td>
<td>185M</td>
</tr>
<tr>
<td>10BASET</td>
<td>twisted-pair</td>
<td>RJ-45</td>
<td>?</td>
</tr>
<tr>
<td>100BASE</td>
<td>fiber/tp</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1000BASE</td>
<td>fiber/copper</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

10BASET, popular, cheap, hub-based, need better grade of wire to support 100 mbit ethernet
10BASE2, daisy chain cable, with T connectors + terminators

Jim Binkley
Enet - properties

- original form: 10 mbps
  - (1.25 mbytes per sec)
- broadcast bus
- distributed access control; i.e., no central “master” saying you may or may not
- hw gets every packet, may not pass it on
- CSMA/CD - carrier sense multiple access with collision detection
enet - rough algorithm

check carrier to see if cable busy (CSMA)
if yes
    wait for idle
else
    transmit and listen for collision (CD)
    if collision
        backoff randomly and try again N times
    else wait min idle time - give others nodes a chance
        (distributed fairness, time slot == 51.2us for 10mbit)
collision detection/retransmission

- N tries, say 16
- if collision, must send jam signal, random backoff and retransmit
- jam == 512 bits (64 bytes), make sure end nodes hear collision, hence enet min frame is 64 bytes (46 data)
- backoff is “binary exponential algorithm”
- wait 1, 2, 4, 8 time-slots, etc * a random delay, max 1023
- packets can be lost due to collision, especially if network is heavily used
- modern network cards can saturate cable;
- best utilization put at %30 (over elapsed time)
ethernet addressing

- each controller has **UNIQUE (!) ethernet or MAC address**, assigned via IEEE in its “brains” (rom, flash memory, whatever)
- 48-bit integer, 6 unsigned char bytes
  - unicast address: **00:00:C0:01:02:03**
- first 3 bytes are manufacturer code
  - Intel: **00:AA:00**
  - Sun: **08:00:20**
- /standards.ieee.org/db/oui/index.html - IEEE web page for MAC lookup

Jim Binkley
3 kinds of physical address

- **unicast** - physical address of controller
- **broadcast**: ff:ff:ff:ff:ff:ff
- **multicast**: 01:xx:xx:xx:xx:xx
- **IP multicast range**: [01:00:5E:00:00:00..01:00:5E:7f:ff:ff]
- **ip-enet mapping not 1-1, 32 ip addr to 1 enet/ip multicast address**
Ethernet frame formats

◆ what does packet look like on wire?
◆ at least two formats
  – IEEE 802.3 (Novell/ISO/some UNIX)
  – Ethernet 2.0 (traditional UNIX/Xerox NS)
◆ 802.3 has 2 sub-layers
  – Logical Link Control - handles demux to net layer
  – Media Access Control - addressing/i/o
## IEEE Data Link Layer (2)

<table>
<thead>
<tr>
<th>LLC - Logical Link Control (IEEE 802.2) - net layer demux, error handling</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAC (media access control) layer</td>
</tr>
<tr>
<td>CSMA/CD IEEE 802.3 (Ethernet)</td>
</tr>
<tr>
<td>Token Bus 802.4 (defunct)</td>
</tr>
<tr>
<td>Token Ring 802.5</td>
</tr>
<tr>
<td>new, 802.6 802.11</td>
</tr>
</tbody>
</table>

MAC - 48 bit IEEE addresses

Jim Binkley
Ethernet 2.0 frame format

min = 64 bytes, max = 1518

<table>
<thead>
<tr>
<th>dst</th>
<th>src</th>
<th>type</th>
<th>data</th>
<th>crc</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>6</td>
<td>2</td>
<td>46-1500</td>
<td>4</td>
</tr>
</tbody>
</table>

ip type = 0x800
arp type = 0x806, 18 bytes of padding (0)
rarp type = 0x8035
802.3 frame format

- min = 64 bytes, max = 1518

<table>
<thead>
<tr>
<th>dst</th>
<th>src</th>
<th>len</th>
<th>llc crud</th>
<th>type</th>
<th>data</th>
<th>crc</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>6</td>
<td>2</td>
<td>6</td>
<td>2</td>
<td>38-1492</td>
<td>4</td>
</tr>
</tbody>
</table>

So how can driver tell difference between 802.3 and E 2.0?

Jim Binkley
and the mystery envelope...

- they don’t overlap. len $\geq 46$ && $\leq 1500$
- ip type == 0x800, 2048 in decimal
headers/trailers

- 8 byte preamble used for synchronization
- CRC is 32 bit “hash code”, if computed crc != packet crc, packet is tossed
- no retries, so-called “best effort”
- what does enet CRC guarantee you?
- what doesn’t it guarantee you?
bad things happen to good pkts

- all bit errors are caught by CRC? (no)
  - ethernet crc is better than IP checksum though
- most are caught? (yes)
- that your packet will arrive for sure? (no)
  - collisions or output i/f may toss as too busy
  - routers are busy and throw packets out (congestion)
  - “noise” causes CRC error, therefore packet is tossed
- if you have 10 routers end to end, CRC is enough to guarantee reliability? (no way)
- where would bad memory hurt a packet?
IP and Modems

- roughly 3 things might be done, focus = #2
  - 1. text-only terminal emulation - dialup
     » kermit, pcplus (procomm), UNIX telnet session
  - 2. link-layer full network access (slip/ppp)
  - 3. application-level tunnel/gateway (linux term)
     » client/server application gateway, client and server communicate directly via rs-232, talk to apps via unix sockets

Jim Binkley
slip/ppp net diagram

- **apps**
  - ftp, www
  - tcp
  - ip
  - slip/ppp
  - serial dev

- **link layer**
  - rs-232
  - modem
  - blink, blink...

- **IP POP** (router only)
  - ftp, www
  - tcp
  - ip
  - slip/ppp
  - serial dev

- **Inet**
oh, btw

- change the names and previous picture describes Internet backbone too...
- modem -> CSU/DSU (say to T1)
- IP boxes on both sides are routers
- connection might be permanent or dynamic (on demand dialup popular with ISDN)
slip - serial line IP

- the “not a standard standard”, RFC 1055
- simple, no protocol header, just one/two byte framing characters around data

Pros
- extremely simple, common

Cons
- can’t support non-ip net layers (ipx) as no header
- no CRC, reliability (modern modems - may not matter)
- can’t negotiate anything (ip address, compression)

Jim Binkley
slip protocol (SIC!)

- data 0xc0, 0xc0 is frame char
- need escape char (if 0xc0 is data?)
  - SLIP ESC = 0xdb, on sending
  - if see 0xc0, substitute 0xdb 0xdc
  - if see 0xdb, substitute 0xdb 0xdd
- CSLIP or Van Jacobson Compression
  - tcp headers only, not udp, not tcp connection
  - not the data!, not ping (icmp on ip)
ppp - point to point protocol

- architecture at link layer has 2 parts
  - *network control part* (NCP), handles demux to network layer, any network options
    » example, for IP, handle dynamic ip addr exchange
  - *link control part* (LCP), handle link management, reliable (better) communication

- plus *encapsulation (frame) with header for pkt*
  - CRC, multi-protocol, framing as features
  - VJ compression but only for tcp headers
## PPP link-layer architecture

<table>
<thead>
<tr>
<th>IP</th>
<th>Net. Control Proto.</th>
<th>Appletalk NCP</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Link Control Protocol (LCP)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Serial Communication Driver -- RS-232, ISDN</td>
<td></td>
</tr>
</tbody>
</table>

Cons: complex to debug (at least compared to slip!)
Pros: IETF protocol used by Novell, Appletalk

Jim Binkley
PPP - rfcs

- rfc 1661 - fundamentals including protocol types for LCP part, state machine, etc.
- 1332 - IP/NCP part
  - address negotiation
  - VJ compression
- CHAP (see radius as well)
- and rfcs for new link-layer technology framing and other more clever bits

Jim Binkley
PPP - a few bullet items

- 16-bit error correction - not as strong as enet
  - possibly duplicated by modem-level protocol?
- multi-protocol; e.g., appletalk/novell/ip
- CHAP - challenge response authentication with shared secret password on both sides as well as PAP which is plaintext password
- client ip address can be dynamically negotiated
- may be used in WAN context as well (ISDN)
- SLIP is mostly extinct

Jim Binkley
**PPP frame format**

- **Header (5 bytes)**
  - **Flag**: 0x7e
  - **Addr**: 0xff
  - **Control**: 0x03
  - **Protocol**: ip = 0x0021

- **Data 0..1500**

- **Trailer (3 bytes)**
  - **CRC (2 bytes)**
  - **Flag**: 0x7e

- **LCP proto, 0xc021, NCP 8021, data x0021**

Jim Binkley
PPP protocol

- protocol roughly consists of:
  - .lcp link establishment and subsequent
    » close and periodic link status check
  - optional lcp link authentication
  - NCP phase
    » e.g., IP address negotiation and/or VJ compression
  - final lcp shutdown
- LCP has a number of packet types, configure, terminate, error, echo, etc.
loopback driver

- special IP address, 127.0.0.1
- everything you write to it, comes back up stack
- “localhost” (DNS) -> 127.0.0.1
- % telnet localhost | 127.0.0.1
- a few controllers can’t read own transmissions, so loopback is useful there too (in addition to preventing unnecessary net traffic)

Jim Binkley
MTU - max transfer unit

- limit on size of frame transmitted at link layer
- on UNIX: \% \texttt{netstat -in} (or \texttt{ifconfig -a}?!)
  - enet II: 1500, 802.3: 1492
  - slip: 1004 (ftp/thruput), 296 (telnet/share)
  - usoft ppp: 1500
  - ATM: around 8-9k, fddi: 4352
- if ip has bigger packet, it \textbf{fragments} the pkt

Jim Binkley
PATH - MTU (avoid fragmentation)

- transport layer determines best link-layer MTU from end to end, RFC 1191 Deering/Mogul
- older and lamentable TCP algorithm:
  - if dst on same subnet
    - send at MTU size (or 1024!)
  - else
    - send at router MSS: 576
- PATH MTU exists in most hosts, but easier for routers to do. host must keep tcp/ip state
  - routers simply send ICMP error message with needed next-link MTU back to source end system, pkts marked Dont Fragment
repeaters/bridges/routers

◆ **repeaters (hubs)** - function at physical layer (l1)
  - active hw device, strengthen signal
  - simply tie wires together, still same net
  - may have sw brains, managed means speaks SNMP
  - may not forward collisions (or it may)

◆ **bridges(switches)** - function at device layer (l2)
  - adaptive/learning bridges isolate same-side traffic
  - must flood broadcasts

◆ **routers** - operate at network layer (l3)

Jim Binkley
Jim Binkley
bridge (or switch? or hub?)

- has **sw** that acts on link layer MAC addresses
- may filter (security) based on MAC address
- network isolation (don’t forward garbage)
- may be adaptive learner (efficient)
- may have spanning tree (redundant)
- may be “switch” (parallel) and speak VLAN
- typically same media (enet) on all ports
  - although cross media bridges exist
traditional bridge operation

- i/fs are in promiscuous mode - read all pkts
- collisions aren’t forwarded THEREFORE
- network isolation which repeaters can’t do (hubs do this)
- learn which packets belong to which side
- bridges as “switches” are rage now
  - fast bus, 10 10mbps enet -> 100 mbit bus
  - support “multimedia”, one node per wire
- bridges have **spanning tree algorithm** with own link-layer protocols, form tree to prevent loops - allows redundancy

Jim Binkley
bridge learning mode

- Look at input’s src MAC address
- If broadcast or multicast, must forward
- If address not in lookup table, store as (address, i/o port, timestamp)
- If address on “new” port, change entry
- If address on “old” port, update timestamp
bridge forwarding algorithm

- if dst address broadcast/multicast forward
- if address in database
  - if input port same as listed port, don’t forward
  - else forward out other port
- else
  - forward (and store!)
bridge (adaptive/learning)

src A to dst B learns to not forward
src A to dst C must always forward

link layer

MAC=A  MAC=B

driver layer

MAC=C

ip net = 200.1.2.x

packets

ip net = 200.1.2.x

Jim Binkley
what’s wrong?

ethernet segment #1

b1

b2

ethernet segment #2

assume 2 bridges hook 2 ethernet segments together. no problem, right?

Jim Binkley
spanning-tree

- see Stallings, Local and Metropolitan Area Networks, for more info
- IEEE 802 standard (802.1D)
- bridge protocol at link layer
- bridges form rooted tree
- leave “cycles” out; i.e., port may be left out of spanning tree and not work (blocked state)
- done with simple link-layer flooding
4 bridges, what happens?
trad. bridge function summary

- adaptive learning - unicast isolation as long as MAC src location can be learned
- same broadcast domain on both sides - forward multicast/broadcast
- store and forward, therefore collision detection (modern switches may not do this as must store to calculate crc)
- spanning tree - prevent link loops
enet switch vs “bridge” or hub

- in a switch, packets forwarded from port A to port B are forwarded in parallel
- in a hub, not so
- switch means fewer collisions if one node per wire as unicast can’t collide (full-duplex means no collisions)
- switch might use “store/forward” (traditional bridge) or “cut through” (switches will be bridges too)
- cut through means pkt only examined up to dst MAC address
- hubs are often repeaters anyway (e.g., 10BASE-T), but do collision detection (bridge function)
bridge as switch

ideal: one port/one node

computer node/hub

Jim Binkley

10/100mbit enet: bridge backplane N * 10/100
bridge/switch considerations

- **broadcast domain** - “segment” over which broadcasts are forwarded and heard
- **collision domain** - “segment” over which collisions can occur

- have to ask ourselves what these mean in terms of switches/bridges/hubs/repeaters?
- switch setup for cut thru cannot detect collisions (need to look at entire packet)

Jim Binkley
level 3/4 - switching/VLAN

- beware the marketroids - some think this is oxymoron (level 7 switching ...)
- VLAN means we have ability in switch to logically group segments
- VLAN X on port Y/Z, means Y/Z have shared broadcast domain.
  - logical ethernet segment, not necessarily physical
- on router/switch, thus if pkt crosses from VLAN Y to X, then only is routed
VLAN picture - combined router/switch

ports: A, B, C, D

- vlan X = ports A/D, pkts to B routed

Jim Binkley
vlans and switches and subnets

- Assume IP subnet 1 to 1 with vlan
- Logical vlan connectivity MAY exist (under negotiation in IEEE)
- Means -- intra and inter switch vlans
- Port i, j on switch I, and port X on switch Y all in same vlan V
- Cisco tag switching is one proprietary example

Jim Binkley
router

network layer

<table>
<thead>
<tr>
<th>network layer/ip</th>
</tr>
</thead>
<tbody>
<tr>
<td>driver layer</td>
</tr>
<tr>
<td>physical</td>
</tr>
</tbody>
</table>

ip net = 200.1.2.x  <->  packets  <->  ip net = 200.1.3.x

Jim Binkley
how does router affect collision/bcast domain?

- broadcasts are NOT usually forwarded
  - exceptions exist: e.g., DHCP/BOOTP request
- multicast the SAME, (barring multicast routing)
- collision domain limited as well
- routers may be viewed as absolute sanity firewalls for ethernet segment disasters
  - broadcast meltdown ...

Jim Binkley
“typical” network topology

- edge router
- some telco tech.
- switch/hub
- 10mbit
- end nodes (TP)
- switch/hub
- fast switch (backplane)
- servers
- hi-speed interconnect to another switch

note: different tech.

Jim Binkley